轻量级认证加密算法 ASCON 的差分功耗分析

潘　力，韦永壮

（桂林电子科技大学 计算机与信息安全学院，广西 桂林　541004）

摘　要：针对轻量级认证加密算法 ASCON 的结构，提出一种差分功耗分析方法。该方法结合算法 S 盒实现特点，利用汉明重量模型作为功耗区分函数，将功耗曲线分组，并恢复出加密用的主密钥。进一步地，对于 DPA 攻击中出现的“魅峰”，给出一种功耗曲线预处理方法，先将曲线根据明文分组并求均值，再对预处理后的曲线发起 DPA 攻击。通过采集 s^a 置换泄露的 1 500 条功耗曲线，能快速恢复出其主密钥的 44 bit。此外，直接攻击原始曲线所需时间为 21 849.888 9 ms，引入预处理技术后，攻击预处理的曲线所需时间为 198.911 3 ms，约为直接攻击原始曲线所需时间的 1/109。

关键词：ASCON 轻量级认证加密算法；差分能量分析；汉明重量模型；预处理

中图分类号：TN918.4　　　**文献标志码**：A　　　**文章编号**：1673-808X(2023)01-0048-07

Differential power analysis of lightweight authenticated encryption algorithm ASCON

PAN Li，WEI Yongzhuang

（School of Computer and Information Security，Guilin University of Electronic Technology，Guilin 541004，China）

Abstract：Aiming at the structure of the lightweight authentication encryption algorithm ASCON，a differential power analysis) method is proposed. It combines the implementation characteristics of the algorithm S-box，uses the Hamming weight model as the power consumption discrimination function，groups the traces，and recovers the master key for encryption. Furthermore，for the "ghost peaks" what appear in DPA attacks，a traces preprocessing method is given. First，the traces are grouped according to plaintext and averaged，and then DPA attacks are launched on the preprocessed traces. The 44 bit master key of ASCON cipher can be recovered by attacking its s^a permutation，where 1 500 traces are collected. In addition，the time required to directly attack the original traces is 21 849.888 9 ms，and the time required to attack the preprocessed traces is 198.911 3 ms. After preprocessing the traces，the time taken to attack the preprocessed traces is about 1/109 of that of directly attacking the original traces.

Key words：ASCON lightweight authenticated encryption algorithm；differential power analysis；Hamming weight model；preprocessing

近些年来，在一些新兴领域（如传感器网络、医疗保健、分布式控制系统、物联网、信息物理系统等）的资源受限设备[1]需要相互连接，协同完成某些工作[2-3]。由于大多数加密算法都是为桌面/服务器环境设计的，这些算法并不适合资源受限设备。2014 年，荷兰拉德堡德大学、奥地利格拉茨技术大学、德国英飞凌科技公司密码专家联合设计出新的轻量级加密算法 ASCON[4]。ASCON 底层基于 320 bit 置换，具有高安全性，同时便于软硬件快速实现。ASCON 的置换定义在 64 bit 机器字上，适合在 64 bit 平台上快速切片实现，而比特交错允许 32/16/8 bit 平台上的快速切片实现。2015 年，美国国家标准技术研究院（national institute of standards and technology，简称 NIST）面向全球启动轻量级加密算法的征集工作。2019 年 4 月 18 日，NIST

收稿日期：2022-03-12
基金项目：国家自然科学基金（61872103，62062026）；广西自然科学基金（2019GXNSFGA245004）。
作者简介：韦永壮（1976－），男，教授，博士，研究方向为对称密码算法设计与分析、加密芯片侧信道攻击与防御等。E-mail：walker_wyz@guet.edu.cn
引文格式：潘力，韦永壮．轻量级认证加密算法 ASCON 的差分功耗分析[J]．桂林电子科技大学学报，2023，43(1)：48-54.

公布了进入第一轮的 56 个算法名单,同年 8 月 30 日,NIST 公布了进入第二轮的 32 个算法名单。2019 年,ASCON 成为 CAESAR(competition for authenticated encryption: security, applicability, and robustness)竞赛 6 个胜选算法之一,同年,ASCON 提交至 NIST 轻量级密码算法征集工作组。2021 年,ASCON 成为 NIST 第三轮 10 个候选算法之一。目前该算法的安全性备受广泛关注。

2015 年,Dobraunig 等[5]给出了 ASCON 线性分析的结果,改进了算法设计者给出的差分分析结果。2017 年,Samwel 等[6]将差分功耗分析(differential power analysis,简称 DPA)[7]应用于 ASCON 算法的硬件实现,通过攻击在 SAKURA-G 开发板上采集到的 50 000 条功耗曲线,恢复出了 64 bit 主密钥。Gross 等[8]评估了 ASCON 的硬件实现开销。2019 年,Ramezanpour 等[9]基于双故障注入和密钥划分技术,提出了一种基于统计无效故障分析(statistical ineffective fault analysis,简称 SIFA)的方法,并应用于 ASCON 算法的软件实现。Bar-on 等[10]提出了差分线性连接表(differential-linear connectivity table,简称 DLCT),允许攻击者考虑 2 个子密钥之间的依赖性,同时证明可以使用快速傅立叶变换有效构建 DLCT,并将其应用于 ASCON 算法的差分线性分析。2020 年,Surya 等[11]提出了一种在 FPGA 上实现的对于 ASCON 算法的时钟故障注入攻击方法。2021 年,Rohit 等[12]给出了 ASCON 算法的 7 轮不同攻击复杂度的 2 种密钥恢复攻击方法。同年,Rohit 等[13]提出了应用于 ASCON 算法的 7 轮密钥恢复攻击,并给出了几种密钥恢复攻击方案,同时进一步改进了 4~6 轮的立方区分器。Joshi 等[14]提出一种称为预攻击(preliminary attack,简称 PA)的密钥恢复攻击方案,并评估了 ASCON 算法对于故障分析攻击的安全性。Basel 等[15]证明了当初始化阶段的轮数减少时,可以使用立方攻击恢复密钥。目前,针对 ASCON 算法的传统数学分析和故障攻击已经取得一些进展,同时对于部署在硬件平台的 ASCON 算法的侧信道攻击也有一些进展。但 ASCON 算法部署在软件平台上时抵御 DPA 攻击的能力还有待进一步评估。

根据 ASCON 加密算法的结构和特点,设计了针对 ASCON 算法的 DPA 攻击方案。引入预处理技术,加快了 DPA 攻击速度。攻击 s^a 置换的实验结果表明,ASCON 算法具备有限的抗 DPA 攻击能力。

1　ASCON 算法描述

Dobraunig 等[4]基于海绵结构设计了 ASCON 加密算法,这是一种轻量级的认证加密算法,算法内部采用基于 SPN 结构的置换。Dobraunig 等[4]提交至 NIST 的 ASCON 算法分为认证加密模式和哈希模式。认证加密模式的 ASCON 分为 ASCON-128、ASCON-128a 和 ASCON-80pq。ASCON-80pq 将密钥长度增加至 160 bit,增强了抵抗量子密钥搜索的能力。为方便起见,此处只讨论 ASCON-128a 加密算法(下文简称 ASCON)的安全性。

ASCON 的内部状态 s 为 320 bit,加密时主要使用 s^a 和 s^b 两种轮数不同的置换。ASCON 算法流程如图 1 所示,整个加密过程包括初始化、关联数据吸收、加密明文 P 和生成标签 t。初始化和生成标签使用安全性较高的 s^a,数据处理则使用安全性稍弱的 s^b。加密过程如下:

1)初始化阶段:初始向量 I、128 bit 密钥 K、128 bit 随机向量 $\boldsymbol{v}$ 进入 s^a,生成初始状态。其中 $I=$ 0xa0400c06,初始状态为 $s_{\text{init}}=I\|K\|\boldsymbol{v}$。

2)数据处理阶段:数据的处理以 128 bit 为一组分组进行,不足 128 bit 的分组用 0 填充。首先吸收关联数据 A,而后加密明文。在数据处理过程中,被处理数据与内部中间状态 s_{in} 的高 128 bit 进行异或运算后,接着 s_{in} 接入 s^b 更新。加密明文时,明文与 s_{in} 异或运算后的结果直接析出为相应的密文。

3)标签生成阶段:数据处理结束后,s_{in} 与 K 异或,进入 s^a 更新。更新后的 s_{in} 后 128 bit 与 K 做异或运算,析出 128 bit 标签 t。

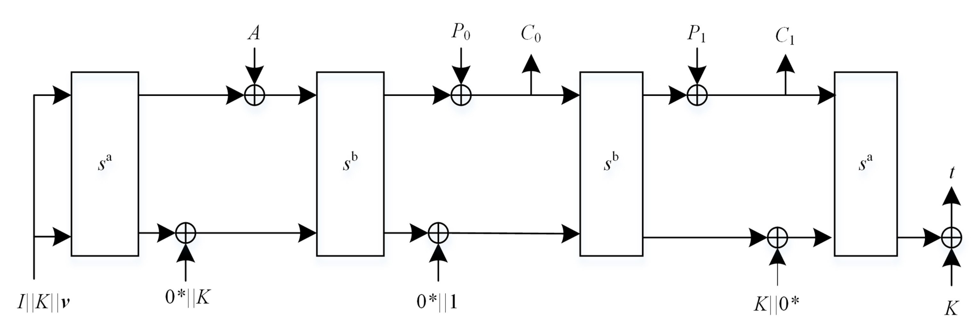


图 1　ASCON 算法加密流程

1.1 置换

ASCON 包含 12 轮的 s^a 和 8 轮的 s^b 两种置换。它们使用相同的 5 bit 密码 S 盒，即 5 bit 输入得到 5 bit 输出，如表 1 所示。

表 1 S 盒

X	$S(X)$	X	$S(X)$
0	4	16	30
1	11	17	19
2	31	18	7
3	20	19	14
4	26	20	0
5	21	21	13
6	9	22	17
7	2	23	24
8	27	24	16
9	5	25	12
10	8	26	1
11	18	27	25
12	29	28	22
13	3	29	10
14	6	30	15
15	28	31	23

ASCON 算法的 S 盒采用切片技术实现，S 盒放置状态如图 2 所示。64 个 S 盒并排放置，其中 x_0 表示高位，x_4 表示低位。

在数据传入 S 盒前，x_2 先与轮常量做异或运算。每轮的轮常量如表 2 所示。

表 2 轮常量

s^a	s^b	轮常量	s^a	s^b	轮常量
0		0xf0	6	2	0x96
1		0xe1	7	3	0x87
2		0xd2	8	4	0x78
3		0xc3	9	5	0x69
4	0	0xb4	10	6	0x5a
5	1	0xa5	11	7	0x4b

1.2 扩散层

扩散层为内部状态 s 的 5 个 64 bit 寄存器字提供扩散。Σ 函数应用于不同寄存器字中，对不同寄存器字使用不同的循环常量。该函数衍生于 SHA-2 算法[16]的 Σ 函数，表示为

$$\begin{cases}\sum_0(x_0)=x_0\oplus(x_0>>>19)\oplus(x_0>>>28),\\ \sum_1(x_1)=x_1\oplus(x_1>>>61)\oplus(x_1>>>39),\\ \sum_2(x_2)=x_2\oplus(x_2>>>1)\oplus(x_2>>>6),\\ \sum_3(x_3)=x_3\oplus(x_3>>>10)\oplus(x_3>>>17),\\ \sum_4(x_4)=x_4\oplus(x_4>>>7)\oplus(x_4>>>41)。\end{cases} \quad (1)$$

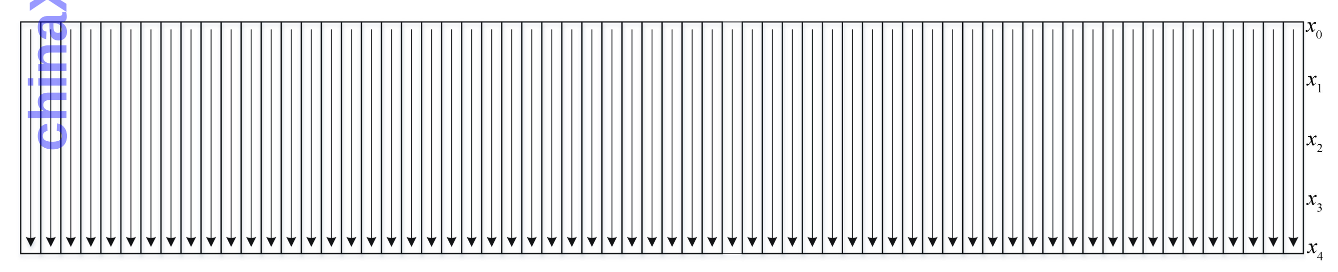


图 2 S 盒放置状态

2 功耗分析

Kocher 等[7]利用密码芯片的功耗与设备当前所处理的中间值（中间状态）的依赖关系，提出了 DPA。Hu 等[17]提出了一种高效 DPA，并研究一个临界区分函数，使密钥恢复更加有效和实用。Sim 等[18]提出了差分分析辅助功耗攻击（differential analysis aided power attack，简称 DAPA）方法，将 DPA 运用在（非）线性移位寄存器上。Chen 等[19]提出了归一化 DPA（bormalized differential power analysis，简称 NDPA），以规避“魅峰”，其性能优于传统的 DPA。DPA 的核心思想是根据猜测密钥和选定的功耗区分函数将采集到的功耗曲线 T 进行分组，并求解出组间均值差的最大值，此最大值对应的猜测密钥就是恢复出的密钥 r。Kocher 最早提出的 DPA 分析模型为

$$\Delta_d=\frac{\sum_{i=1}^{n}f(P_i,g_s)T_i(j)}{\sum_{i=1}^{n}f(P_i,g_s)}-\frac{\sum_{i=1}^{n}(1-f(P_i,g_s))T_i(j)}{\sum_{i=1}^{n}(1-f(P_i,g_s))}\approx$$

$$2\left(\frac{\sum_{i=1}^{n} f(P_i, g_s) T_i(j)}{\sum_{i=1}^{n} f(P_i, g_s) T_i} - \frac{\sum_{i=1}^{n} T_i(j)}{n}\right), \tag{2}$$

其中:$f(P_i, g_s)$为功耗区分函数;P_i 为输入的明文或者密文;g_s 为猜测密钥;i 为功耗曲线数;j 为功耗曲线中的采样点。DPA 的攻击流程如图 3 所示。

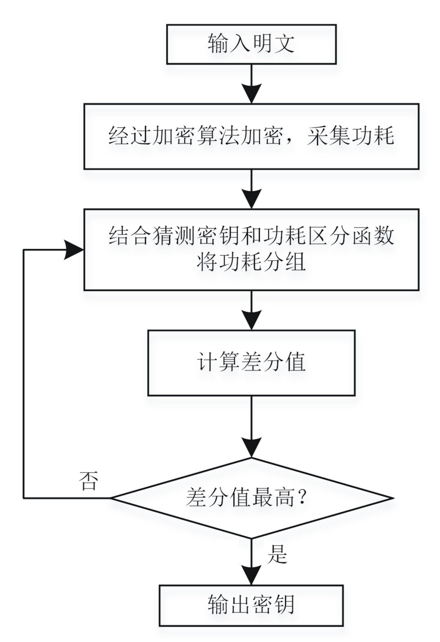


图 3　DPA 攻击流程

2.1　功耗模型

对于在软件平台实现的加密算法,一般选用汉明重量模型刻画动态能量消耗。原因在于,微控制器采用预充电总线,在向总线发送被操作数之前,所有总线都会被置为 1[7]。设某时刻 j 的功耗 $T_i(j)$ 与中间值 $x(j)$的线性关系[7]为

$$T_i(j) = \mu \mathrm{H}(x(j)) + c + n, \tag{3}$$

其中:μ 为汉明重量 $\mathrm{H}(x(j))$与功耗 $T_i(j)$ 之间的常量比率;c 为电路中的静态能量消耗;n 为随机噪声。

2.2　ASCON 算法的功耗分析

为了减少攻击时的计算复杂度,针对 ASCON 算法的攻击,将攻击点选择在初始化阶段第一轮 S 盒输出的位置,如图 4 所示。在进入扩散层前,密钥不会被扩散到其它位置,有利于发起攻击。

经典 DPA 根据功耗区分函数输出值的某比特将功耗曲线分成 2 组,不仅与设备的泄露特性有关,对功耗的利用率也不足。功耗函数定义为 $\mathrm{H}(S(I \| K \| v))$,将汉明重量函数作为功耗区分函数,可以降低对设备泄露特性的依赖,提高成功率。

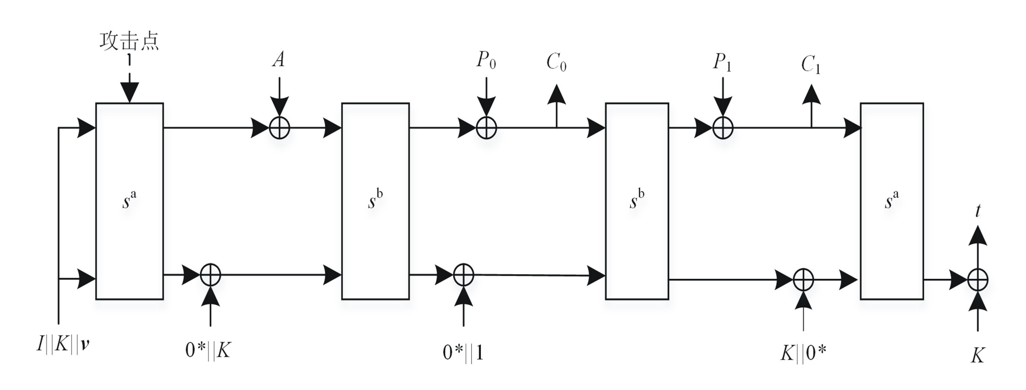


图 4　含攻击点的 ASCON 算法加密流程

在实际分析中,可能存在错误密钥对应的尖峰大于正确密钥对应的尖峰现象,即“魅峰”,导致密钥恢复出错。根据式(3)可知,引起尖峰的可能原因是电路以及外界环境引起的随机噪声。对功耗曲线做预处理,具体步骤如下:

1)从采集到的数据中读取出明文和功耗曲线;

2)对于单个明文,以明文为下标,将功耗曲线根据明文划分成不同组;

3)分组完成后,求出每组的均值,得到与明文对应的均值功耗曲线。此时,功耗曲线所在分组的下标即可视为明文;

4)对于多个明文,重复步骤 2)和 3)。

算法 1　功耗曲线预处理

输入:n 条原始功耗曲线 T,每条曲线 m 个点。n 行明文 P,每行明文有 k 个数。

输出:2^l 条预处理后的功耗曲线 T_{P},每条曲线 m 个点。

for num =0 to k do

```
    for row =0 to n do
      for line =0 to m do
        T_P[k][P[num][row]][line]+=(T[row][line]-T_P[k][P[num][row]][line])/tracesnum[P[num][row]]
        tracesnum[P[num][row]]++
  end
  return T_P
```

其中，l 为进入 S 盒时单个明文的长度。由式(3)可知，当被加密的明文相同时，产生的功耗除随机噪声外，其余部分功耗保持不变。曲线预处理的思想即是将功耗曲线 T 根据对应的明文分成 2^l 组，并计算每组的均值。预处理后，功耗曲线的条数从 n 变为 2^l 组，此时功耗曲线对应的下标即为明文。DPA 攻击一般需要大量功耗曲线，即 $n \gg 2^l$，预处理后加速了分析速度。

对预处理后的功耗曲线 T_P 发起攻击，此时曲线下标即为明文，具体步骤如下：

1)读取出预处理后的功耗曲线 T_P；

2)对于单个明文，根据功耗区分函数，明文和猜测密钥得到的汉明重量小于($l/2+0.5$)时，将与明文相对应的功耗曲线划分为一组，剩余曲线划分为另一组；

3)计算两组功耗曲线的组内均值，再求解出组间均值差，并记录猜测密钥和最大差值；

4)遍历猜测密钥并计算出每个猜测密钥对应的最大差值。当差值达到最大时，此差值所对应的猜测密钥即为恢复出的密钥；

5)对于多个明文，重复步骤 2)～5)。

算法 2 DPA 攻击

输入：2^l 条功耗曲线 T_P，每条曲线 m 个点。2^l 行明文 P，每行明文 k 个数据。

输出：恢复出的 k 个密钥 r。

```
for num =0 to k do
  for row =0 to 2^l do
    for g =0 to 2^l do
      flag =H(S(P+g)) <(l/2+0.5) ? 0:1
      for line =0 to m do
        traces[flag][line] +=(T_P[row][line]-traces[flag][line])/num[flag][row]
        num[flag][row]++
  for line =0 to m do
    D[line] = | traces[0][line]-traces[1][line] |
    if max < D[line]
      max =D[line]
      r[k] =g
end
return r
```

3 测试结果与分析

实验环境如表 3 所示。

表 3 实验环境

设备	型号
CPU	Xeon E5-2643 v4
内存	32 GiB
开发板	STM32F407
示波器	PicoScope 3206D

在实验中，ASCON 算法用 C 语言实现，示波器的采样频率为 500 MHz。执行加密算法时，关联数据 A 和明文 P 为固定的 256 bit 数，K 为固定的 128 bit 数，具体参数为：$A=$0x00…00，$P=$0x00…00，$K=$ 0x000102030405060708090a0b0c0d0e0f。同时改变随机向量 $\boldsymbol{v}$，并采集相应的功耗曲线，结果如图 5 所示。

图 5 中，每条曲线包含 12 300 个点，采集 1 500 条功耗曲线，并执行 DPA，恢复出的密钥 r 为 0x00000000001000080819bc1c196a1950。从结果可以看出，总共恢复出了 44 bit 密钥。原因在于，S 盒采用切片技术实现时，算法使用 64 个并行运算的 S

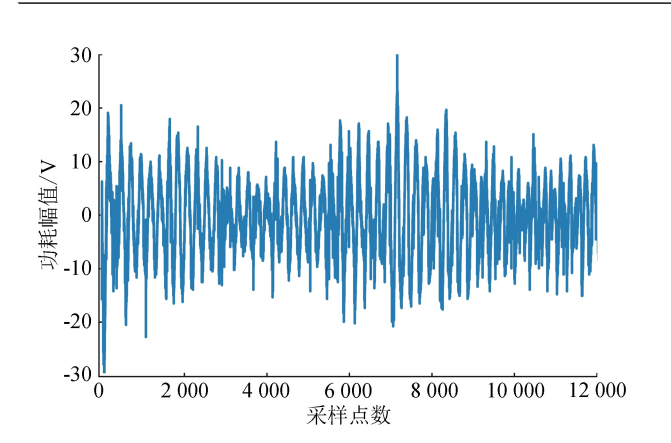


图 5　ASCON 功耗

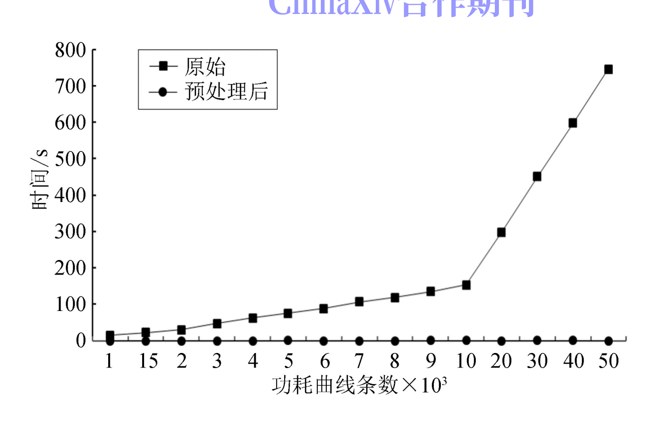


图 6　时间复杂度

盒,这 64 个 S 盒的功耗会彼此影响,采用经典 DPA 分析,效率较为低下。

此外,将本研究与文献[6]、[8]中的 ASCON 算法在不同平台上的实现方案以及文献[20]中具有类似结构的 WAGE 算法在攻击复杂度方面进行对比,结果如表 4 所示。

表 4　攻击复杂度对比

算法	实现平台	功耗函数/bit	功耗曲线/×10³ 条
本研究 ASCON	STM32F407	5	1.5
ASCON[6]	SAKURA-G	5	50
ASCON[8]	仿真	5	5
WAGE[20]	Cortex-M4F	7	10

文献[6]和文献[8]给出了 ASCON 算法采用硬件实现时的攻击结果。文献[6]中,采集部署在 SAKURA-G 开发板上的 ASCON 算法的 50 000 条真实功耗曲线,通过攻击恢复了 64 bit 密钥信息。文献[8]对 ASCON 算法的硬件实现方案进行了仿真,得到了相应的模拟功耗曲线,并完全恢复了加密用的 128 bit 主密钥。这些模拟功耗曲线没有环境噪声和采样噪声,更有利于发起 DPA 攻击。文献[20]分析了 NIST 第二轮候选算法之一 WAGE,利用采集的 10 000 条功耗曲线恢复出 128 bit 密钥信息。通过对部署在 STM32 开发平台的 ASCON 算法发起攻击,恢复出 44 bit 密钥信息。由于软硬件平台的功耗表现不同,此处对 ASCON 算法在软件平台的抗 DPA 能力做了评估。

为了验证预处理方案对 DPA 攻击的加速效果,采集 50 000 条功耗曲线,直接攻击原始功耗曲线和攻击预处理后的功耗曲线的时间复杂度如图 6 所示。

通过调用 C 语言提供的库函数可计算出攻击所用的时间。其中,攻击 1 500 条原始曲线所需时间约为 21 849.888 9 ms,攻击 1 500 条预处理后的曲线约为 198.911 3 ms。从图 6 可看出,随着功耗曲线的增多,直接攻击原始曲线所需的时间逐渐增长,而攻击预处理后的功耗曲线所需时间几乎不变。原因在于,原始功耗曲线经预处理后,曲线条数始终为 2^5 条,即 32 条。因此,攻击预处理后的曲线所需时间几乎不变,极大地加速了攻击过程。

4　结束语

分析了 ASCON 算法抵御 DPA 攻击的能力,即主要分析了 s^a 置换对于 DPA 攻击的安全性。在 ARM 开发板上的实验结果表明,采用 DPA 攻击部件 s^a 泄露的 1 500 条功耗曲线,可恢复 44 bit 主密钥,即 ASCON 算法存在敏感信息泄露。下一步将研究 ASCON 算法的高效掩码防护方法。

参考文献:

[1] 向宏,夏晓峰. 轻量级密码在资源受限设备安全中的应用简析[J]. 自动化博览,2018,35(增刊 2):72-75.

[2] KHAN M A,SALAH K. IoT security:review,blockchain solutions,and open challenges[J]. Future Generation Computer Systems,2018,82:395-411.

[3] MAHMUD R,KOTAGIRI R,BUYYA R. Internet of Everything[M]. Berlin,Heidelberg:Springer Press,2018:103-130.

[4] DOBRAUNIG C,EICHLSEDER M,Mendel F,et al. Ascon[EB/OL]. (2021-4-14)[2021-11-23]. https://csrc. nist. gov/projects/lightweight-cryptography/finalists.

[5] DOBRAUNIG C,EICHLSEDER M,MENDEL F,et al. Cryptanalysis of ascon[C]//Topics in Cryptology. Berlin,Heidelberg:Springer Press,2015:371-387.

[6] SAMWEL N,DAEMEN J. DPA on hardware implementations of ascon and keyak[C]//Computing Frontiers Conference. New York:ACM Press,2017:415-24.

[7] MANGARD S, OSWALD E, POPP T. Power Analysis Attacks-revealing the Secrets of Smart Cards[M]. Berlin, Heidelberg: Springer, 2010: 55.

[8] GROSS H, WENGER E, DOBRAUNIG C, et al. Ascon hardware implementations and side-channel evaluation [J]. Microprocessors and Microsystems, 2017(52): 470-479.

[9] RAMEZANPOUR K, AMPADU P, DIEHL W. A statistical fault analysis methodology for the Ascon authenticated cipher[C]//2019 IEEE International Symposium on Hardware Oriented Security and Trust. Piscataway, NJ: IEEE Press, 2019: 41-50.

[10] BAR-ON A, DUNKELMAN O, KELLER N, et al. DLCT: a new tool for differential-linear cryptanalysis [C]//Advances in Cryptology. Berlin, Heidelberg: Springer, 2019: 313-342.

[11] SURYA G, MAISTRI P, SANKARAN S. Local clock glitching fault injection with application to the ASCON cipher[C]//2020 IEEE International Symposium on Smart Electronic Systems. Piscataway, NJ: IEEE Press, 2020: 271-276.

[12] ROHIT R, SARKAR S. Diving deep into the weak keys of round reduced Ascon[J]. IACR Transactions on Symmetric Cryptology, 2021(4): 74-99.

[13] ROHIT R, HU K, SARKAR S, et al. Misuse-free key-recovery and distinguishing attacks on 7-round Ascon [J]. IACR Transactions on Symmetric Cryptology, 2021(3): 102-136.

[14] JOSHI P, MAZUMDAR B. SSFA: subset fault analysis of ASCON-128 authenticated cipher[J]. Microelectronics Reliability, 2021: 114-155.

[15] BASEL H, JORGE D S. Hardware Supply Chain Security[M]. Berlin, Heidelberg: Springer Press, 2021: 69-88.

[16] QUYNH H D. Secure hash standard[EB/OL]. (2020-2-27) [2021-12-23]. http://dx.doi.org/10.6028/NIST.FIPS.180-4.

[17] HU Q S, FAN X N, ZHANG Q W. An effective differential power attack method for advanced encryption standard[C]//2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. Piscataway, NJ: IEEE Press, 2019: 58-61.

[18] SIM S M, JAP D, BHASIN S. DAPA: Differential analysis aided power attack on (non) linearfeedback shift registers[C]//Cryptographic Hardware and Embedded Systems. Berlin, Heidelberg: Springer Press, 2020: 169-191.

[19] CHEN J C, NG J-S, KYAW N A, et al. Normalized differential power analysis for ghost peaks mitigation [C]//2021 IEEE International Symposium on Circuits and Systems. Piscataway, NJ: IEEE Press, 2021: 1-5.

[20] FEI Y S, GONG G, CHENG G Y, et al. Correlation power analysis and higher-order masking implementation of WAGE[C]//Selected Areas in Cryptography. Berlin, Heidelberg: Springer Press, 2020: 593-14.

编辑：张所滨